Target: Beagle Bone Black I2C1 Controller

Chapter 7

Review SCL and SDA signals.

Start Condition

Stop Condition

ACK

High-level initialization

Inspect P9 connector of the Beagle Bone Black for

Identify the I2C1\_SCL pin

Find the Control Module register to change Pin17 to the I2C1\_SCL signal.

Identify the I2C1\_SDA pin

Find the Control Module register to change Pin18 to the I2C1\_SDA signal.

The display module should already be physically connected already.

Initialize the Clock Module for I2C1

Study I2C section of the Sitara manual

Analyze 7-bit addressing mode

Follow high-level list of steps in the section to initialize the I2C controller.

Find the settings to reach 12 MHz clock

Find the settings to get a 100 kbps SCL for standard mode operation (F/S).

Create an initialization pin map list for the required registers.

Registers for initialization

Registers for transmission start

Registers for byte transfer when controller is ready

Construct the High-level algorithm for initialization and transmission start.

Should be the same as the chapter 7 example, minus the slave read.

Construct the Low-level algorithm initialization and transmission start.

Study the manual to determine how to get it in the desired mode.

Determine how you send characters to display

Make a list of desired initialization words

Make a list of words needed to display your name.

Either single height on two lines, or double height display mode.

Complete the High-level and low-level algorithms with these steps included.

Create a polled version for handshaking of the program.

Receive TA/Instructor sign-off

Modify Algorithm to implement the handshaking on an interrupt basis with the interrupt controller.

Modify Program to implement handshaking on an interrupt basis.

Receive TA/Instructor Sign-off

Find how to make the display rotate around the screen to the right in a loop. Or blink on and off.

**Chapter 7**

**Review SCL and SDA signals.**

* Both lines driven by open drain or open collector transistors.
  + The lines require pull up resistors because of this.
* Serial Data (SDA)
* Serial Clock Line (SCL)

Start Condition

* At start of transmission (if the bus is available and that a master wants to send a message to a slave) the master pulls the SDA line from high to low, while the SCL line is high.
* The SCL line is then pulsed, shifting out the data bits on SDA synchronously with the SCL pulses.
  + The most significant bit is shifted out first on the SDA line.
* If the slave receives 8 bits correctly, it synchronously pulls the SDA line low as an acknowledge signal to the master
* The SCL line can be held low by the slave if time is needed to process the byte, forcing the master to insert wait states.
* When the slave releases the SCL line and it is pulled high by the external pull-up resistor. The master can then send another byte.
* If no acknowledge signal after a byte is generated, the master can either generate a stop condition on the bus to abort/end the transfer or assert a repeated start condition on the bus to start a new transmission.

Stop Condition

* The master allows the SDA line to transition high while the SCL line is high.
* For a repeated start condition, the SDA line is pulled low while the SCL line is high.
* Either stop condition option is available after any transmission.

ACK

* The bit following each byte transmitted.
* The master will signal the end of the transmission to the slave by not pulling the acknowledge bit low for the last byte that was clocked out of the slave.

Data transfer

* For any transmission on the bus, the master will first send out an address byte.
* The upper 7 bits of the first byte sent out by the master will contain the address of the slave that is to be written to or read from.
* The least significant bit of this byte will be a 0 for a write operation and a 1 for a read operation.
* After the master receives an acknowledge signal from the addressed slave, it then clocks out the data byte of the message.

High-level initialization

Inspect P9 connector of the Beagle Bone Black for

Identify the I2C1\_SCL pin

Find the Control Module register to change Pin17 to the I2C1\_SCL signal.

Identify the I2C1\_SDA pin

Find the Control Module register to change Pin18 to the I2C1\_SDA signal.

The display module should already be physically connected already.

Initialize the Clock Module for I2C1

Study I2C section of the Sitara manual

Analyze 7-bit addressing mode

Follow high-level list of steps in the section to initialize the I2C controller.

Find the settings to reach 12 MHz clock

Find the settings to get a 100 kbps SCL for standard mode operation (F/S).

* I2C Low-speed operation is 0 to 100KHz
* I2C Fast bus operation is 0 to 400KHz
* I2C High-speed operation is 0 to 3.4 MHz

Create an initialization pin map list for the required registers.

Registers for initialization

Registers for transmission start

Registers for byte transfer when controller is ready

Construct the High-level algorithm for initialization and transmission start.

Should be the same as the chapter 7 example, minus the slave read.

Construct the Low-level algorithm initialization and transmission start.

Study the manual to determine how to get it in the desired mode.

Determine how you send characters to display

Make a list of desired initialization words

Make a list of words needed to display your name.

Either single height on two lines, or double height display mode.

Complete the High-level and low-level algorithms with these steps included.

Create a polled version for handshaking of the program.

Receive TA/Instructor sign-off

Modify Algorithm to implement the handshaking on an interrupt basis with the interrupt controller.

Modify Program to implement handshaking on an interrupt basis.

Receive TA/Instructor Sign-off

Find how to make the display rotate around the screen to the right in a loop. Or blink on and off.